ALPHACERTIFIED: CERTIFYING SOLUTIONS TO POLYNOMIAL SYSTEMS

JONATHAN D. HAUENSTEIN AND FRANK SOTTILE

Abstract. Smale's α-theory uses estimates related to the convergence of Newton's method to certify that Newton iterations will converge quadratically to solutions to a square polynomial system. The program **alphaCertified** implements algorithms based on α-theory to certify solutions of polynomial systems using both exact rational arithmetic and arbitrary precision floating point arithmetic. It also implements algorithms that certify whether a given point corresponds to a real solution, and algorithms to heuristically validate solutions to overdetermined systems. Examples are presented to demonstrate the algorithms.

Introduction

Current implementations of numerical homotopy algorithms [1, 32, 38] such as PHCpack [41], HOM4PS [27], Bertini [4], and NAG4M2 [28] routinely and reliably solve systems of polynomial equations with dozens of variables having thousands of solutions. Here, 'solve' means 'compute numerical approximations to solutions.' In each of these software packages, the solutions are validated heuristically—often by monitoring iterations of Newton's method. This works well in practice, giving solutions that are acceptable in most applications. However, a well-known shortcoming of numerical methods for computing approximate solutions to systems of polynomials is that the output is not certified. This restricts their use in some applications, including those in pure mathematics. The program **alphaCertified** is intended to remedy this shortcoming.

In the 1980's, Smale [36] and others investigated the convergence of Newton's method, developing α-theory [9, Ch. 8]. This refers to a computable positive constant $\alpha(f,x)$ depending upon a system $f\colon \mathbb{C}^n \to \mathbb{C}^n$ of polynomials and a point $x \in \mathbb{C}^n$ such that, if

$$\alpha(f,x) \;<\; \frac{13-3\sqrt{17}}{4} \;\approx\; 0.157671\,,$$

then iterations of Newton's method starting at x will converge quadratically to a solution to f, which is a point $\xi \in \mathbb{C}^n$ with $f(\xi)=0$. In principle, Smale's α-theory provides certificates for validating numerical computations with polynomials.

1991 *Mathematics Subject Classification.* 65G20, 65H05.

Key words and phrases. certified solutions, alpha theory, polynomial system, numerical algebraic geometry.

Research of Hauenstein supported in part by the Fields Institute and NSF grant DMS-1114336.

Research of both authors supported in part by NSF grants DMS-0915211 and DMS-0922866, and the Royal Swedish Academy of Sciences through the Institut Mittag-Leffler.

Current implementations of numerical homotopy algorithms do not incorporate α-theory to certify their output or their path-tracking. There have been two projects which use fixed double precision and focus on certified path-tracking. Malajovich [30] released the most recent version of his Polynomial System Solver in 2003, which uses α-theory to certify toric path-tracking algorithms, but he states that "[it] is actually not intended for an end user." Beltrán and Leykin [8] have recently shown how to use α-theory to certify path-tracking, and hence the output of numerical homotopy algorithms. While they demonstrate that certification can dramatically affect the speed of computation, this is an important development, as certified path-tracking is necessary for applications such as numerical irreducible decomposition [37] or computing Galois groups [29]. They are continuing this line of research.

We describe a program, **alphaCertified**, that implements elements of α-theory to certify numerical solutions to systems of polynomial equations using both exact rational and arbitrary precision floating point arithmetic. As it only certifies the output of a numerical computation, it avoids the bottlenecks of certified tracking, while delivering some of its benefits. Given a square polynomial system $f : \mathbb{C}^n \to \mathbb{C}^n$, **alphaCertified** uses Smale's α-theory to answer the following three questions for a finite set of points $X \subset \mathbb{C}^n$:

(1) From which points of X will Newton's method converge quadratically to some solution to f?
(2) From which points of X will Newton's method converge quadratically to distinct solutions to f?
(3) If f is real ($\{\overline{f_1}, \ldots, \overline{f_n}\} = \{f_1, \ldots, f_n\}$), from which points of X will Newton's method converge quadratically to real solutions to f?

Often, a sharp upper bound B on the number of roots to a square polynomial system f is known. Given a set of B points, **alphaCertified** can be used to certify that iterations of Newton's method starting from each point in the set converge quadratically to some solution to f and that these solutions are distinct. This guarantees that each of the B roots of f can be approximated to arbitrary accuracy using Newton's method. Moreover, **alphaCertified** can certify how many of the B solutions to f are real when f is real.

A polynomial system $f : \mathbb{C}^n \to \mathbb{C}^N$ is *overdetermined* if $N > n$, that is, if the number of polynomials exceeds the number of variables. Dedieu and Shub [12] studied Newton's method for overdetermined polynomial systems and gave conditions which guarantee quadratic convergence of its iterations. Unlike square systems, the fixed points of this overdetermined Newton's method need not be solutions. For example, $x = 1$ is a fixed point of Newton's method applied to $f(x) = \begin{bmatrix} x \\ x-2 \end{bmatrix}$.

The program **alphaCertified** validates solutions to overdetermined systems. Given a finite set $X \subset \mathbb{C}^n$ and an overdetermined system, it generates two or more random square subsystems, answers the three questions above for each, and compares the results. In particular, given $\delta > 0$, it can certify that, for a given approximate solution to two or more random subsystems, the associated solutions all lie within a distance δ of each other. For a given δ, this heuristically validates solutions to overdetermined systems.

In summary, **alphaCertified** is novel in each of the following ways. It implements algorithms from α-theory using either exact rational or arbitrary precision floating point arithmetic. When using exact rational arithmetic with a square polynomial system, its implementation of α-theory is completely rigorous. It certifiably determines if an approximate solution corresponds to a real solution, which may be used to count the real solutions to a polynomial system, and it uses α-theory to obtain information on the roots of overdetermined systems. The examples we give demonstrate the practicality of certification based on α-theory, and its viability as an alternative to exact symbolic methods, as the certificates for square systems when using exact rational arithmetic are mathematical proofs of computed results.

In Section 1, we review the concepts of α-theory utilized by **alphaCertified**. Section 2 presents the algorithms for square polynomial systems while Section 3 describes our approach to overdetermined polynomial systems. Implementation details are presented in Section 4 with examples presented in Section 5 verifying some computational results in kinematics and generating evidence for conjectures in enumerative real algebraic geometry.

1. Smale's α-theory

We summarize key points of Smale's α-theory for square polynomial systems that are utilized by **alphaCertified**. More details may be found in [9, Ch. 8].

Let $f : \mathbb{C}^n \to \mathbb{C}^n$ be a system of n polynomials in n variables with common zeroes $\mathcal{V}(f) := \{\xi \in \mathbb{C}^n \mid f(\xi) = 0\}$, and let $Df(x)$ be the Jacobian matrix of the system f at x. Consider the map $N_f : \mathbb{C}^n \to \mathbb{C}^n$ defined by

$$N_f(x) \ := \ \begin{cases} x - Df(x)^{-1}f(x) & \text{if } Df(x) \text{ is invertible,} \\ x & \text{otherwise.} \end{cases}$$

The point $N_f(x)$ is called the *Newton iteration of f starting at x*. For $k \in \mathbb{N}$, let

$$N_f^k(x) \ := \ \underbrace{N_f \circ \cdots \circ N_f(x)}_{k \text{ times}}$$

be the k^{th} Newton iteration of f starting at x.

Definition 1. Let $f : \mathbb{C}^n \to \mathbb{C}^n$ be a polynomial system. A point $x \in \mathbb{C}^n$ is an *approximate solution* to f with *associated solution* $\xi \in \mathcal{V}(f)$ if, for every $k \in \mathbb{N}$,

$$\|N_f^k(x) - \xi\| \ \leq \ \left(\frac{1}{2}\right)^{2^k-1} \|x - \xi\| \,. \tag{1}$$

That is, the sequence $\{N_f^k(x) \mid k \in \mathbb{N}\}$ *converges quadratically* to ξ. Here, $\|\cdot\|$ is the usual hermitian norm on $\mathbb{C}^n$, namely $\|(x_1, \ldots, x_n)\| = (|x_1|^2 + \cdots + |x_n|^2)^{1/2}$.

Smale's α-theory describes conditions that imply a given point x is an approximate solution to f. It is based on constants $\alpha(f,x)$, $\beta(f,x)$, and $\gamma(f,x)$. If $Df(x)$ is invertible,

these are

$$\begin{aligned}
\alpha(f,x) &:= \beta(f,x)\gamma(f,x)\,,\\
\beta(f,x) &:= \|x-N_f(x)\| \;=\; \|Df(x)^{-1}f(x)\|\,, \qquad \text{and}\\
\gamma(f,x) &:= \sup_{k\ge 2}\left\|\frac{Df(x)^{-1}D^kf(x)}{k!}\right\|^{\frac{1}{k-1}}\,. \qquad (2)
\end{aligned}$$

If $x\in\mathcal{V}(f)$ is such that $Df(x)$ is not invertible, then we define $\alpha(f,x):=\beta(f,x):=0$ and $\gamma(f,x):=\infty$. Otherwise, if $x\notin\mathcal{V}(f)$ and $Df(x)$ is not invertible, then we define $\alpha(f,x):=\beta(f,x):=\gamma(f,x):=\infty$.

In the formula (2) for $\gamma(f,x)$, the k^{th} derivative $D^kf(x)$ [26, Chap. 5] to f is the symmetric tensor whose components are the partial derivatives of f of order k. It is a linear map from the k-fold symmetric power $S^k\mathbb{C}^n$ of $\mathbb{C}^n$ to $\mathbb{C}^n$. The norm in (2) is the operator norm of $Df(x)^{-1}D^kf(x)\colon S^k\mathbb{C}^n\to\mathbb{C}^n$, defined with respect to the norm on $S^k\mathbb{C}^n$ that is dual to the standard unitarily invariant norm on homogeneous polynomials [25],

$$\Big\|\sum_{|\nu|=d} a_\nu x^\nu\Big\|^2 \;:=\; \sum_{|\nu|=d}|a_\nu|^2/\tbinom{d}{\nu}\,,$$

where $\nu=(\nu_1,\ldots,\nu_n)$ is an exponent vector of non-negative integers with $x^\nu=x_1^{\nu_1}\cdots x_n^{\nu_n}$, $|\nu|=\nu_1+\cdots+\nu_n$, and $\binom{d}{\nu}=\frac{d!}{\nu_1!\cdots\nu_n!}$ is the multinomial coefficient.

The following version of Theorem 2 from page 160 of [9] provides a certificate that a point x is an approximate solution to f.

Theorem 2. *If $f:\mathbb{C}^n\to\mathbb{C}^n$ is a polynomial system and $x\in\mathbb{C}^n$ with*

$$\alpha(f,x) \;<\; \frac{13-3\sqrt{17}}{4} \;\approx\; 0.157671\,, \qquad (3)$$

then x is an approximate solution to f. Additionally, $\|x-\xi\|\le 2\beta(f,x)$ where $\xi\in\mathcal{V}(f)$ is the associated solution to x.

Remark 3. If $\alpha(f,x)\ge\frac14$, then x may not be an approximate solution to f. For example, for $f(x)=x^2$, if $x\neq 0$, then x is not an approximate solution to f yet $\alpha(f,x)=\frac14$.

For a polynomial system $f:\mathbb{C}^n\to\mathbb{C}^n$ and a point $x\in\mathbb{C}^n$, we say that x is a *certified approximate solution* to f if (3) holds.

Theorem 4 and Remark 6 of [9, Ch. 8] give a version of Theorem 2 that **alphaCertified** uses to certify that two approximate solutions have the same associated solution.

Theorem 4. *Let $f:\mathbb{C}^n\to\mathbb{C}^n$ be a polynomial system, $x\in\mathbb{C}^n$ with $\alpha(f,x)<0.03$ and $\xi\in\mathcal{V}(f)$ the associated solution to x. If $y\in\mathbb{C}^n$ with*

$$\|x-y\| \;<\; \frac{1}{20\gamma(f,x)}\,,$$

then y is an approximate solution to f with associated solution ξ.

1.1. **Bounding higher order derivatives.** The constant $\gamma(f,x)$ encoding the behavior of the higher order derivatives of f at x is difficult to compute, but it can be bounded above. For a polynomial $g:\mathbb{C}^n\to\mathbb{C}$ of degree d, say $g=\sum_{|\nu|\leq d} a_\nu x^\nu$, define

$$\|g\|^2 \ := \ \sum_{|\nu|\leq d} |a_\nu|^2 \, \frac{\nu!(d-|\nu|)!}{d!}\,.$$

Then $\|\cdot\|$ is the standard unitarily invariant norm on the homogenization of g. For a polynomial system $f:\mathbb{C}^n\to\mathbb{C}^n$, define

$$\|f\|^2 \ := \ \sum_{i=1}^n \|f_i\|^2 \qquad \text{where} \qquad f(x) \ = \ \left[\begin{array}{c} f_1(x) \\ \vdots \\ f_n(x) \end{array}\right],$$

and for a point $x\in\mathbb{C}^n$, define

$$\|x\|_1^2 \ := \ 1+\|x\|^2 \ = \ 1+\sum_{i=1}^n |x_i|^2\,.$$

Let $\Delta_{(d)}(x)$ be the $n\times n$ diagonal matrix with

$$\Delta_{(d)}(x)_{i,i} \ := \ d_i^{1/2}\|x\|_1^{d_i-1}\,,$$

where d_i is the degree of f_i. If $Df(x)$ is invertible, define

$$\mu(f,x) \ := \ \max\{1, \|f\|\cdot\|Df(x)^{-1}\Delta_{(d)}(x)\|\}\,.$$

The following version of Proposition 3 from §I-3 of [35] gives an upper bound for $\gamma(f,x)$.

Proposition 5. *Let $f:\mathbb{C}^n\to\mathbb{C}^n$ be a polynomial system with $d_i=\deg f_i$ and $D=\max d_i$. If $x\in\mathbb{C}^n$ such that $Df(x)$ is invertible, then*

$$\gamma(f,x) \ \leq \ \frac{\mu(f,x)D^{\frac{3}{2}}}{2\|x\|_1}\,. \tag{4}$$

2. Algorithms for square polynomial systems

Let $f:\mathbb{C}^n\to\mathbb{C}^n$ be a square polynomial system and $X=\{x_1,\ldots,x_k\}\subset\mathbb{C}^n$ be a set of points. We describe the algorithms implemented in **alphaCertified** which answer the three questions posed in the Introduction. These algorithms are stated for a polynomial system with complex coefficients, but are implemented for polynomial systems with coefficients in $\mathbb{Q}[\sqrt{-1}]$ using both exact and arbitrary precision arithmetic.

For each $i=1,\ldots,k$, **alphaCertified** first checks if $f(x_i)=0$. If $f(x_i)\neq 0$, then **alphaCertified** determines if $Df(x_i)$ is invertible. If it is, **alphaCertified** computes $\beta(f,x_i)$ and upper bounds for $\alpha(f,x_i)$ and $\gamma(f,x_i)$ using the following algorithm.

Procedure $(\alpha,\beta,\gamma)=$ **ComputeConstants**(f,x)**:**
Input: A square polynomial system $f:\mathbb{C}^n\to\mathbb{C}^n$ and a point $x\in\mathbb{C}^n$ such that $Df(x)$ is invertible.
Output: $\alpha:=\beta\cdot\gamma$, $\beta:=\|Df(x)^{-1}f(x)\|$, and γ, where γ is the upper bound for $\gamma(f,x)$ given in Proposition 5.

The next algorithm uses Theorem 2 to compute a subset Y of X containing points that are certified approximate solutions to f.

Procedure $Y =$ **CertifySolns**(f, X)**:**
Input: A square polynomial system $f : \mathbb{C}^n \to \mathbb{C}^n$ and a set $X = \{x_1, \ldots, x_k\} \subset \mathbb{C}^n$.
Output: A set $Y \subset X$ of approximate solutions to f.
Begin:

(1) Initialize $Y := \{\}$.
(2) For $j = 1, 2, \ldots, k$, if $f(x_j) = 0$, set $Y := Y \cup \{x_j\}$, otherwise, do the following if $Df(x_j)$ is invertible:
   (a) Set $(\alpha, \beta, \gamma) :=$ **ComputeConstants**(f, x_j).
   (b) If $\alpha < \dfrac{13 - 3\sqrt{17}}{4}$, set $Y := Y \cup \{x_j\}$.

Return: Y

As **alphaCertified** uses the upper bound for $\gamma(f, x)$ of Proposition 5, it may fail to certify a legitimate approximate solution x to f. In that case, a user may consider retrying after applying a few Newton iterations to x. The software **alphaCertified** does not invoke an automatic refinement to inputs that it does not certify. This is because Newton iterations may have unpredictable behavior (attracting cycles, chaos) when applied to points that are not in a basin of attraction. However, **alphaCertified** does provide the functionality for the user to do this refinement.

Suppose that x is an approximate solution to f with associated solution ξ such that $Df(\xi)$ is invertible. Since x is an approximate solution, $\beta(f, N_f^k(x))$ converges to zero. Since $\gamma(f, x)$ is the supremum of a finite number of continuous functions of x, $\gamma(f, N_f^k(x))$ is bounded. In particular, $\alpha(f, N_f^k(x))$ converges to zero.

Given approximate solutions x_1 and x_2 to f with associated solutions ξ_1 and ξ_2, respectively, Theorems 2 and 4 can be used to determine if ξ_1 and ξ_2 are equal. In particular, if

$$\|x_1 - x_2\| \; > \; 2(\beta(f, x_1) + \beta(f, x_2))\,,$$

then $\xi_1 \neq \xi_2$ by Theorem 2. If on the other hand we have

$$\alpha(f, x_i) \; < \; 0.03 \qquad \text{and} \qquad \|x_1 - x_2\| \; < \; \frac{1}{20\gamma(f, x_i)}$$

for either $i = 1$ or $i = 2$, then $\xi_1 = \xi_2$ by Theorem 4. This justifies the following algorithm which determines if two approximate solutions correspond to distinct associated solutions.

Procedure $\mathit{isDistinct} =$ **CertifyDistinctSoln**(f, x_1, x_2)**:**
Input: A square polynomial system $f : \mathbb{C}^n \to \mathbb{C}^n$ and approximate solutions x_1 and x_2 to f with associated solutions ξ_1 and ξ_2, respectively, such that $Df(\xi_1)$ and $Df(\xi_2)$ are invertible.
Output: A boolean $\mathit{isDistinct}$ that describes if $\xi_1 \neq \xi_2$.
Begin: Do the following:

(a) For $i = 1, 2$, set $(\alpha_i, \beta_i, \gamma_i) :=$ **ComputeConstants**(f, x_i).
(b) If $\|x_1 - x_2\| > 2(\beta_1 + \beta_2)$, **Return** True.
(c) If $\alpha_i < 0.03$ and $\|x_1 - x_2\| < \dfrac{1}{20\gamma_i}$, for either $i = 1$ or $i = 2$, **Return** False.

(d) For $i = 1, 2$, update $x_i := N_f(x_i)$ and return to (a).

This will halt, determining whether or not $\xi_1 = \xi_2$ as $\beta(f, N_f^k(x_i))$ decreases quadratically with k, while $\gamma(f, N_f^k(x_i))$ is bounded.

2.1. **Certifying real solutions.** A polynomial system $f : \mathbb{C}^n \to \mathbb{C}^n$ is *real* if $\{\overline{f_1}, \ldots, \overline{f_n}\} = \{f_1, \ldots, f_n\}$. In that case, solutions to $f(x) = 0$ are either real or occur in conjugate pairs. Also, $N_f(\overline{x}) = \overline{N_f(x)}$ for $x \in \mathbb{C}^n$ so that $N_f \colon \mathbb{R}^n \to \mathbb{R}^n$ is a real map. Theorems 2 and 4 can be used to determine if an approximate solution of f is associated to a real solution. Let x be an approximate solution to f with associated solution ξ. We do not assume that x is real, for numerical continuation solvers yield complex approximate solutions. By assumption, $\overline{x}$ is also an approximate solution to f with associated solution $\overline{\xi}$. If

$$\|x - \overline{x}\| \;>\; 2\left(\beta(f, x) + \beta(f, \overline{x})\right) \;=\; 4\beta(f, x)\,,$$

then $\xi \neq \overline{\xi}$ by Theorem 2 since

$$\|\xi - \overline{\xi}\| \;\geq\; \|x - \overline{x}\| - 4\beta(f, x) \;>\; 0\,.$$

Consider the natural projection map $\pi_{\mathbb{R}} : \mathbb{C}^n \to \mathbb{R}^n$ defined by

$$\pi_{\mathbb{R}}(x) \;=\; \frac{x + \overline{x}}{2}\,.$$

Since $\|x - \overline{x}\| = 2\|x - \pi_{\mathbb{R}}(x)\|$, ξ is not real if

$$\|x - \pi_{\mathbb{R}}(x)\| \;>\; 2\beta(f, x)\,. \tag{5}$$

We have both a local and a global approach to show that ξ is real. For the local approach, Theorem 4 implies that $\pi_{\mathbb{R}}(x)$ is also an approximate solution to f with associated solution ξ if

$$\alpha(f, x) \;<\; 0.03 \qquad \text{and} \qquad \|x - \pi_{\mathbb{R}}(x)\| \;<\; \frac{1}{20\gamma(f, x)}\,. \tag{6}$$

Since N_f is a real map and $\pi_{\mathbb{R}}(x) \in \mathbb{R}^n$, this implies that $\xi \in \mathbb{R}^n$.

We could also have showed that both x and $\overline{x}$ correspond to the same solution to deduce that $\xi = \overline{\xi}$. If

$$\alpha(f, x) \;<\; 0.03 \qquad \text{and} \qquad \|x - \overline{x}\| \;<\; \frac{1}{20\gamma(f, x)}\,,$$

then Theorem 4 implies that $\xi = \overline{\xi}$. This is more restrictive then (6) since $\|x - \overline{x}\| = 2\|x - \pi_{\mathbb{R}}(x)\|$.

When $\alpha(f, x) <\; 0.03$, (5) and (6) yield closely related statements. Since

$$\frac{5}{3}\beta(f, x) \;=\; \frac{5\alpha(f, x)}{3\gamma(f, x)} \;<\; \frac{5 \cdot 0.03}{3\gamma(f, x)} \;=\; \frac{1}{20\gamma(f, x)}\,,$$

we know that ξ is real if $\|x - \pi_{\mathbb{R}}(x)\| \leq \frac{5}{3}\beta(f, x)$ and not real if $\|x - \pi_{\mathbb{R}}(x)\| > 2\beta(f, x)$.

The following algorithm uses the local approach of (5) and (6) to determine if an approximate solution corresponds to a real associated solution.

Procedure *isReal* = **CertifyRealSoln**(f, x)**:**

Input: A real square polynomial system $f : \mathbb{C}^n \to \mathbb{C}^n$ and an approximate solution $x \in \mathbb{C}^n$ with associated solution ξ such that $Df(\xi)$ is invertible.
Output: A boolean *isReal* that describes if $\xi \in \mathbb{R}^n$.
Begin: Do the following:
(a) Set $(\alpha, \beta, \gamma) :=$ **ComputeConstants**(f, x).
(b) If $\|x - \pi_{\mathbb{R}}(x)\| > 2\beta$, **Return** False.
(c) If $\alpha < 0.03$ and $\|x - \pi_{\mathbb{R}}(x)\| < \dfrac{1}{20\gamma}$, **Return** True.
(d) Update $x := N_f(x)$, and return to (a).

For the global approach to certifying real solutions, suppose that we know *a priori* that f has exactly k solutions. Suppose that $x_1, \ldots, x_k$ are approximate solutions of f with distinct associated solutions. If, for all $j \neq i$, $\overline{x_i}$ and x_j also correspond to distinct solutions, then x_i and $\overline{x_i}$ must correspond to the same solution, which is therefore real. This global approach requires *a priori* knowledge about $\mathcal{V}(f)$ as well as approximate solutions corresponding to each solution to f. While it cannot be applied to all systems, it is an alternative to the test based on $\gamma(f, x)$.

2.2. **Certification algorithm.** For a given set of points X and a polynomial system f, **CertifySolns**, **CertifyDistinctSoln**, and **CertifyRealSoln** answer the three questions posed in the Introduction. We provide a sketch of the algorithm.

Procedure $(A, D, R) =$ **CertifyCount**(f, X)**:**
Input: A square polynomial system $f : \mathbb{C}^n \to \mathbb{C}^n$ and a finite set of points $X = \{x_1, \ldots, x_\ell\} \subset \mathbb{C}^n$ such that if x_j is an approximate solution with associated solution ξ_j, then $Df(\xi_j)$ is invertible.
Output: A set $A \subset X$ consisting of certified approximate solutions to f, a set $D \subset A$ consisting of points which have distinct associated solutions, and, if f is a real map, a subset $R \subset D$ consisting of points which have real associated solutions.
Begin:
(1) Set $A :=$ **CertifySolns**(f, X).
(2) Set $n_A := |A|$ and enumerate the points in A as $a_1, \ldots, a_{n_A}$.
(3) For $j = 1, \ldots, n_A$, set $s_j :=$ *True*.
(4) For $j = 1, \ldots, n_A$ and for $k = j+1, \ldots, n_A$, if s_j and s_k are *True*, set $s_k :=$ **CertifyDistinctSoln**(f, a_j, a_k).
(5) Set $D := \{a_j \mid s_j = \textit{True}\}$.
(6) Initialize $R := \{\}$.
(7) If f is a real polynomial system, do the following:
  (a) Set $n_D := |D|$ and enumerate the points in D as $d_1, \ldots, d_{n_D}$.
  (b) For $j = 1, \ldots, n_D$, if **CertifyRealSoln**(f, d_j) is *True*, update $R := R \cup \{d_j\}$.

3. Overdetermined polynomial systems

When $N > n$, the polynomial system $f : \mathbb{C}^n \to \mathbb{C}^N$ is overdetermined. Dedieu and Shub [12] studied the overdetermined Newton's method whose iterates are defined by

$$N_f(x) \ := \ x - Df(x)^\dagger f(x)\,, \tag{7}$$

where $Df^\dagger(x)$ is the Moore-Penrose pseudoinverse of $Df(x)$ [17, § 5.5.4] to determine conditions that guarantee quadratic convergence. Since the fixed points of N_f may not be solutions to the overdetermined polynomial system f, this approach cannot certify solutions to overdetermined polynomial systems.

We instead certify that points are associated solutions to two or more random square subsystems using the algorithms of Section 2. An additional level of security may be added by certifying that, for a given point which is an approximate solution to two or more random square subsystems, the associated solutions lie within a given distance of each other. As with the overdetermined Newton's method (7), this also cannot certify solutions to overdetermined polynomial systems, which is still an open problem.

Let $R\colon \mathbb{C}^N \to \mathbb{C}^n$ be a linear map, considered as a matrix in $\mathbb{C}^{n\times N}$. Then $\mathcal{R}(f)(x) = R\circ f(x)$ gives a square polynomial system $\mathcal{R}(f) : \mathbb{C}^n \to \mathbb{C}^n$. Since $\mathcal{V}(f) \subset \mathcal{V}(\mathcal{R}(f))$ for any R, we call $\mathcal{R}(f)$ a *square subsystem* of f. There is a nonempty Zariski open subset $\mathcal{A} \subset \mathbb{C}^{n\times N}$ such that for every $R \in \mathcal{A}$ and every $x \in \mathcal{V}(f)$, null $Df(x) = \{0\}$ if and only if $D\mathcal{R}(f)(x)$ is invertible. Moreover, for every $x \in \mathcal{V}(\mathcal{R}(f)) \setminus \mathcal{V}(f)$, $D\mathcal{R}(f)(x)$ is invertible. See [38] for more on square subsystems $\mathcal{R}(f)$.

Define $L = \{f(x) \mid x \in \mathbb{C}^n\} \subset \mathbb{C}^N$ which has dimension at most n possibly passing through the origin. A dimension-counting argument yields that there is a nonemtpy Zariski open set $\mathcal{B} \subset \mathcal{A}\times\mathcal{A} \subset \mathbb{C}^{n\times N}\times\mathbb{C}^{n\times N}$ such that, for every $(R_1, R_2) \in \mathcal{B}$, $K =$ null $R_1 \cap$ null $R_2 \subset \mathbb{C}^N$ is a linear space of dimension $\max\{N - 2n, 0\}$ passing through the origin and $K \cap L \subset \{0\}$. In particular, if $\mathcal{R}_i(f) = R_i \circ f$, then

$$\mathcal{V}(\mathcal{R}_1(f)) \cap \mathcal{V}(\mathcal{R}_2(f)) \ = \ \mathcal{V}(f)\,.$$

In addition, suppose that x is an approximate solution to both $\mathcal{R}_1(f)$ and $\mathcal{R}_2(f)$ with associated solutions ξ_1 and ξ_2, respectively. For $k \in \mathbb{N}$, define $x_{i,k} = N^k_{\mathcal{R}_i(f)}(x)$ for $i = 1, 2$. If $\xi_1 \neq \xi_2$, there exists $k \in \mathbb{N}$ such that

$$\|x_{1,k} - x_{2,k}\| \ > \ 2(\beta(\mathcal{R}_1(f), x_{1,k}) + \beta(\mathcal{R}_2(f), x_{2,k}))\,,$$

certifying that $\|\xi_1 - \xi_2\| > 0$.

If $\xi_1 = \xi_2$, then, for any $\delta > 0$, there exists $k \in \mathbb{N}$ such that

$$\|x_{1,k} - x_{2,k}\| + 2(\beta(\mathcal{R}_1(f), x_{1,k}) + \beta(\mathcal{R}_2(f), x_{2,k})) \ < \ \delta \tag{8}$$

certifying that $\|\xi_1 - \xi_2\| < \delta$. In particular, this certifies that the solutions ξ_1 and ξ_2 to $\mathcal{R}_1$ and $\mathcal{R}_2$ associated to the common approximate solution x lie within a distance δ of each other. For $\delta \ll 1$, this *heuristically* shows that $\xi_1 = \xi_2$.

In summary, if x is a certified approximate solution to two different square subsystems with distinct associated solutions, a certificate can be produced demonstrating this fact. Also (but not conversely), for any given tolerance $\delta > 0$, a certificate can be produced

that the distance between the associated solutions to the two square subsystems is smaller than δ.

An additional test using the function residual could be added to this process. The following lemma describes such a test.

Lemma 6. *Let $f : \mathbb{C}^n \to \mathbb{C}^N$ be an overdetermined polynomial system, $R \in \mathbb{C}^{n\times N}$, and x be an approximate solution to $\mathcal{R}(f) := R \circ f$ with associated solution ξ such that $\alpha(\mathcal{R}(f), x) \le 0.0125$. Then there exists $\epsilon > 0$ such that if there exists $y \in \mathbb{C}^n$ satisfying*

$$\|x - y\| \;\le\; \frac{1}{40\gamma(\mathcal{R}(f), x)} \qquad \textit{and} \qquad \|f(y)\| \;<\; \epsilon\,,$$

then $\xi \in \mathcal{V}(f)$.

Proof. Define $\nu = \dfrac{1}{40\gamma(\mathcal{R}(f), x)}$ and $B(x, \nu) = \{y \in \mathbb{C}^N \mid \|x - y\| \le \nu\}$. We note that if $\gamma(\mathcal{R}(f), x) = \infty$, since $B(x, \nu) = \{x\}$, it is easy to verify that we can take

$$\epsilon = \begin{cases} 1 & \text{if } f(x) = 0, \\ \frac{\|f(x)\|}{2} & \text{otherwise.} \end{cases}$$

Hence, we can assume that $\gamma(\mathcal{R}(f), x) < \infty$. Since

$$\|x - \xi\| \;\le\; 2\beta(\mathcal{R}(f), x) \;=\; \frac{2\alpha(\mathcal{R}(f), x)}{\gamma(\mathcal{R}(f), x)} \;\le\; \frac{0.025}{\gamma(\mathcal{R}(f), x)} \;=\; \nu\,,$$

$\xi \in B(x, \nu)$. Moreover, Theorem 4 yields that $B(x, \nu) \cap \mathcal{V}(\mathcal{R}(f)) = \{\xi\}$.

Assume $\xi \notin \mathcal{V}(f)$. Since $\mathcal{V}(f) \subset \mathcal{V}(\mathcal{R}(f))$, $B(x, \nu) \cap \mathcal{V}(f) = \emptyset$. In particular, $g(z) = \|f(z)\|$ is positive on the compact set $B(x, \nu)$. Thus, there exists $\epsilon > 0$ such that $\|f(y)\| \ge \epsilon$ for all $y \in B(x, \nu)$. $\square$

Remark 7. For Lemma 6 to give an algorithm, we would need a general bound for the minimum of a positive polynomial on a disk. In cases when such a bound is known, e.g., [24], it is too small to be practical.

4. Implementation details for **alphaCertified**

The program **alphaCertified** is written in C and depends upon GMP [19] and MPFR [14] libraries to perform exact rational and arbitrary precision floating point arithmetic. When using rational arithmetic, all internal computations are *certifiable*. Because of the bit length growth of rational numbers under algebraic computations, **alphaCertified** allows the user to select a precision and use floating point arithmetic in that precision to facilitate computations. Since floating point errors from internal computations are not fully controlled, **alphaCertified** only yields a *soft certificate* when using the floating point arithmetic option. When the polynomial system is overdetermined, **alphaCertified** displays a message informing the user about what it has actually computed.

Three input files are needed to run **alphaCertified**. These files contain the polynomial system, the list of points to test, and the user-defined settings. See [22] for more details regarding exact syntax of these files. The polynomial system is assumed to have rational complex coefficients and described in the input file with respect to the basis of monomials.

That is, the user inputs the coefficient and the exponent of each variable for each monomial term in each polynomial of the polynomial system.

The set of points to test are assumed to have either rational coordinates if using rational arithmetic or floating point coordinates if using floating point arithmetic. When using floating point arithmetic, the points are inputted in the precision selected by the user.

The list of user-defined settings includes the choice between rational and floating point arithmetic, the floating point precision to use for the basic computations if using floating point arithmetic, and which certification algorithm to run. The user can also define a value, say $\tau > 0$, such that, for each certified approximate solution, the associated solution will be approximated to within $10^{-\tau}$ and printed to a file.

The specific output of **alphaCertified** depends upon the user-defined settings. In each case, an on-screen table summarizes the output as well as a file that contains a human-readable summary for each point. The other files created are machine-readable files that can be used in additional computations.

Linear solving operations are performed using an LU decomposition and the spectral matrix norm is bounded above using the Frobenius norm. This choice further worsens the approximation of γ described in Proposition 5, which has two direct consequences on the performance of the algorithms. First, this requires that the value of β must be smaller in order to certify approximate solutions in **CertifySolns**. Second, algorithms **CertifyDistinctSoln** and **CertifyRealSoln** may need to utilize extra Newton iterations. Apart from the added computational cost, the use of GMP and MPFR allows **alphaCertified** to still perform these computations even when using such an approximation of γ.

When using rational arithmetic, **alphaCertified** avoids taking square roots when testing the required inequalities. When using floating point arithmetic, as an effort to control the floating point errors, the internal working precision is increased when updating the point via a Newton iteration, for instance in Step (d) of **CertifyDistinctSoln** and Step (d) of **CertifyRealSoln**.

The software **alphaCertified** determines if a square polynomial system f in n variables is real using two tests. The first test determines if the coefficients of f are real. The second selects a pseudo-random point $y \in \mathbb{Q}^n$ and determines if $\{f_1(y), \ldots, f_n(y)\} = \{\overline{f_1(y)}, \ldots, \overline{f_n(y)}\}$.

The user either instructs **alphaCertified** to bypass all tests and declare that f is real, or which tests to use. If all tests fail, then **alphaCertified** bypasses the real certification. Otherwise, for each approximate solution x with associated solution ξ, **alphaCertified** determines if there exists a real approximate solution that also corresponds to ξ. If the user incorrectly identified f as real, then ξ may not be real. Therefore, **alphaCertified** displays a message informing the user about what it actually has certified.

For an overdetermined polynomial system f, **alphaCertified** only checks to see if all of the coefficients of f are real. In this case, **alphaCertified** randomizes f using real matrices to obtain real square subsystems.

5. Computational examples

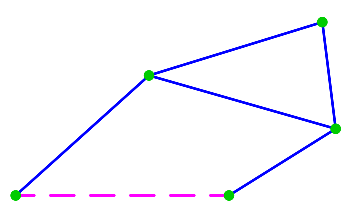
We used **alphaCertified** to study four polynomial systems whose number of real solutions is relevant. Two are from kinematics and two are from enumerative geometry. All involve polynomial systems that are not easily solved using certified methods from symbolic computation. The files used in the computations, as well as instructions for their use, are found on our website [22]. Computations of Sections 5.3 and 5.4 used nodes of the Brazos cluster [10] that consist of two 2.5 GHz Intel Xeon E5420 quad-core processors.

5.1. **Stewart-Gough platform.** The Stewart-Gough platform is a parallel manipulator in which six variable-length actuators are attached between a fixed frame (the ground) and a moving frame (the platform) [18, 40]. Each position of the platform uniquely determines the lengths of the six actuators. However, the lengths of the actuators do not uniquely determine the position and orientation of the platform, as there are typically several assembly modes, called *positions*.

A generic platform with generic actuator lengths has 40 complex assembly modes. Dietmaier [13] used a continuation method to find a platform and leg lengths for which all 40 positions are real. While his formulation as a system of polynomial equations and conclusions about their solutions being real have been reproduced numerically (this is a problem in Verschelde's test suite [42]), these computations only give a heuristic verification of Dietmaier's result.

We modified the polynomial system from Verschelde's test suite, which uses the parameters obtained by Dietmaier, by converting the floating point numbers to rational numbers. We then ran PHCpack [41] on the resulting polynomial system to obtain 40 numerical solutions to the system, each of which it identified as real. After converting the floating point coordinates of the solutions to rational numbers, we ran **alphaCertified** using these rational polynomials and rational points. It verified that these 40 points correspond to distinct real solutions. This gives a rigorous mathematical proof of Dietmaier's result.

5.2. **Four-bar linkages.** A *four-bar linkage* is a planar linkage consisting of a triangle with two of its vertices connected to two bars, whose other endpoints are fixed in the plane. The base of the triangle, the two attached bars, and the implied bar between the two fixed points are the four bars.



A general linkage has a one-dimensional constrained motion during which the joints may rotate, and the curve traced by the apex of the triangle is its *workspace curve*.

The *nine-point path synthesis problem* asks for the four-bar linkages whose workspace curve contains nine given points. Morgan, Sommese, and Wampler [33] used homotopy continuation to solve a polynomial system which describes the four-bar linkages whose workspace curves pass through nine given points. They found that for nine points $\mathcal{P} =$

$\{P_0, \ldots, P_8\} \subset \mathbb{C}^2$ in general position, there are 8652 isolated solutions. Due to a two-fold symmetry, there are 4326 distinct four-bar linkages which appear in 1442 triplets, called Roberts cognates. We used **alphaCertified** to produce a soft certificate that the polynomial system has at least 8652 isolated solutions and, for a specific set of nine real points, certified the number of real solutions among these 8652 solutions.

If $\mathcal{P} \subset \mathbb{R}^2$, the formulation of [33] is not a real polynomial system. The usual approach of writing the variables using real and imaginary parts gives a real polynomial system $f_{\mathcal{P}}$ consisting of four quadratic and eight quartic polynomials. For nine points $\mathcal{P} = \{P_0, \ldots, P_8\} \subset \mathbb{C}^2$, the polynomial system $f_{\mathcal{P}} : \mathbb{C}^{12} \to \mathbb{C}^{12}$ depends upon the variables

$$\{a_1,\, a_2,\, n_1,\, n_2,\, x_1,\, x_2,\, b_1,\, b_2,\, m_1,\, m_2,\, y_1,\, y_2\}\,.$$

Define the complex numbers

$$\begin{array}{lll} a = a_1 + \sqrt{-1}\cdot a_2, & n = n_1 + \sqrt{-1}\cdot n_2, & x = x_1 + \sqrt{-1}\cdot x_2, \\ b = b_1 + \sqrt{-1}\cdot b_2, & m = m_1 + \sqrt{-1}\cdot m_2, & y = y_1 + \sqrt{-1}\cdot y_2, \end{array}$$

whose complex conjugates are $\overline{a}, \overline{n}, \overline{x}, \overline{b}, \overline{m}, \overline{y}$, respectively. These correspond to the variables used in the formulation in [33]. The four quadratic polynomials of $f_{\mathcal{P}}$ are

$$\begin{array}{rclrcl} f_1 &=& n_1 - a_1x_1 - a_2x_2\,, & f_2 &=& n_2 + a_1x_2 - a_2x_1\,, \\ f_3 &=& m_1 - b_1y_1 - b_2y_2\,, & f_4 &=& m_2 + b_1y_2 - b_2y_1\,. \end{array}$$

The eight quartic polynomials depend upon the displacements from P_0 to the other points P_j. For $j = 1, \ldots, 8$, define $Q_j := (Q_{j,1}, Q_{j,2}) = P_j - P_0$ and write each displacement Q_j using *isotropic coordinates*, namely $(\delta_j, \overline{\delta}_j)$ where

$$\delta_j = Q_{j,1} + \sqrt{-1}\cdot Q_{j,2} \ \text{ and } \ \overline{\delta}_j = Q_{j,1} - \sqrt{-1}\cdot Q_{j,2}.$$

For $j = 1, \ldots, 8$, the quartic polynomial f_{4+j} of $f_{\mathcal{P}}$ is

$$f_{4+j} \ := \ \gamma_j\overline{\gamma}_j + \gamma_j\gamma_j^0 + \overline{\gamma}_j\gamma_j^0$$

where

$$\gamma_j := q_j^x r_j^y - q_j^y r_j^x, \qquad \overline{\gamma}_j := r_j^x p_j^y - r_j^y p_j^x, \qquad \gamma_j^0 := p_j^x q_j^y - p_j^y q_j^x$$

and

$$\begin{array}{lll} p_j^x := \overline{n} - \overline{\delta}_j x, & q_j^x := n - \delta_j\overline{x}, & r_j^x := \delta_j(\overline{a} - \overline{x}) + \overline{\delta}_j(a - x) - \delta_j\overline{\delta}_j, \\ p_j^y := \overline{m} - \overline{\delta}_j y, & q_j^y := m - \delta_j\overline{y}, & r_j^y := \delta_j(\overline{b} - \overline{y}) + \overline{\delta}_j(b - y) - \delta_j\overline{\delta}_j. \end{array}$$

We first certified that, for nine randomly selected points in the complex plane, the resulting polynomial system has at least 8652 isolated solutions. Since the displacements Q_j define the polynomial system, we choose them to be points of $\mathbb{Q}[\sqrt{-1}]^2$ with each coordinate having unit modulus of the form

$$\frac{t^2-1}{t^2+1} \ + \ \sqrt{-1}\cdot\frac{2t}{t^2+1}$$

where t was a quotient of two ten digit random integers. We used regeneration [21] in Bertini [4] to compute 8652 points that were *heuristically* within 10^{-100} of an isolated

solution for $f_{\mathcal{P}}$. Then, **alphaCertified** produced a soft certificate using 256-bit precision that these 8652 points are approximate solutions to $f_{\mathcal{P}}$ with distinct associated solutions.

We next certified the number of real solutions for a specific set of nine real points, namely Problem 3 of [33]. The nine real points are listed in Table 2 of [33], which, for convenience, we list the values of δ_j in Table 1. Since the points are real, $\overline{\delta}_j$ is the conjugate

TABLE 1. Values of δ_j for Problem 3 of [33]

j	δ_j
1	$0.27 + 0.1\sqrt{-1}$
2	$0.55 + 0.7\sqrt{-1}$
3	$0.95 + \sqrt{-1}$
4	$1.15 + 1.3\sqrt{-1}$
5	$0.85 + 1.48\sqrt{-1}$
6	$0.45 + 1.4\sqrt{-1}$
7	$-0.05 + \sqrt{-1}$
8	$-0.23 + 0.4\sqrt{-1}$

of δ_j. We used parameter continuation in Bertini to solve the resulting polynomial system starting from the 8652 solutions to the polynomial system solved in the first test. This generated a list of 8652 points which **alphaCertified** soft certified using 256-bit precision to be approximate solutions that have distinct associated solutions of which 384 are real. In particular, this confirms the results reported in Table 3 of [33] for Problem 3, namely, that $64 = 384/6$ of the 1442 mechanisms are real.

Figure 1 shows three of the 64 real mechanisms that solve this synthesis problem, together with their workspace curves. The first has two assembly modes with the workspace

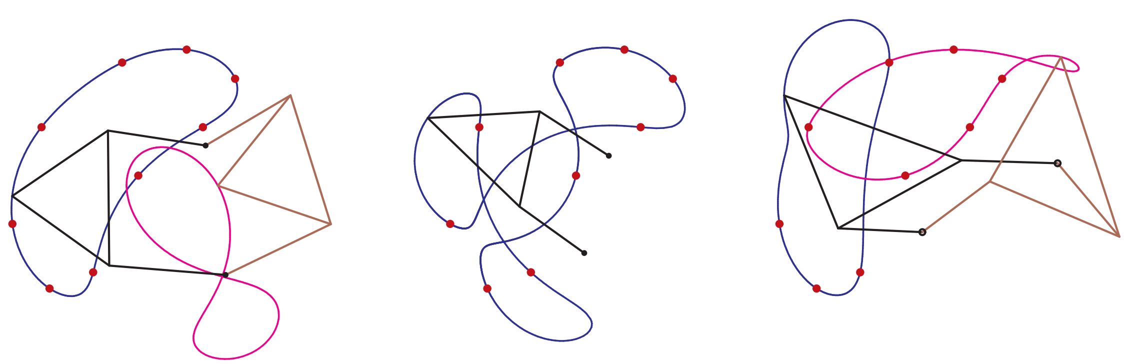


FIGURE 1. Three solutions.

curve of one mode a simple closed curve that contains the nine target points. This mechanism is the only viable mechanism among the 64 real mechanisms. The second has only one assembly mode, but its workspace curve is convoluted and does not meet the target points in a useful order. The third has two assembly modes, and each only reaches a proper subset of the target points.

5.3. **Lines, points, and conics.** We consider geometric problems of plane conics in $\mathbb{C}^3$ that meet k points and $8-2k$ lines for $k = 0, \ldots, 4$. When the points and lines are general, the numbers of plane conics are known and presented in Table 2.

TABLE 2. Numbers of plane conics

k	4	3	2	1	0
Number of conics	0	1	4	18	92

This problem is from a class of problems in enumerative geometry—counting rational curves—that has been of great interest in recent years [15]. For problems of enumerating rational curves of degree d in the plane that interpolate $3d-1$ real points, Welschinger [43] defined an invariant W_d which is a lower bound on the number of real rational curves, and work of Mikhalkin [31] and of Itenberg, Kharlamov, and Shustin showed that W_d is positive and eventually found a formula for it [23].

We used **alphaCertified** to investigate the possible numbers of real solutions to these problems of conics when their input data (points and lines) are real. Of particular interest is the minimum number of solutions that are real. Our experimental data suggests that when $k = 1$ at least two of the solutions will be real, and it shows that for $k = 0, 2$, it is possible to have no real solutions.

This experiment computed random instances of the problem. The coordinates of points were taken to be the quotient of two random ten digit integers, and the real lines were taken to be lines through two such points. The resulting polynomial system was square. Each real instance was solved by Bertini [4] using a straight line parameter homotopy starting with a fixed random complex instance (see [38] for more details). This gave points that were *heuristically* within 10^{-75} of each isolated solution. Then **alphaCertified** used 256-bit precision to softly certify that the points computed by Bertini were approximate solutions corresponding to distinct solutions, and to count the number of real solutions. Since enumerative geometry provides the generic root count, this yields a post-processing certificate that Bertini has indeed computed an approximate solution corresponding to each solution to the polynomial system.

In every instance that Bertini successfully tracked every path, the heuristic results of Bertini matched the certified results of **alphaCertified**. Out of the over 1,450,000,000 paths tracked, 76 paths were truncated by Bertini due to a fail-safe measure. Thirty-two paths were truncated since they needed more than the fail-safe limit of 10,000 steps along the path. Each of these paths were successfully tracked when the limit was raised to 25,000 steps. Forty-four paths were truncated since the adaptive precision tracking algorithm [5, 6, 3] requested to use more than the fail-safe limit of 1024-bit precision. Each of these paths were successfully tracked when the fail-safe limit was raised to 1284-bit precision.

The first interesting case is when $k = 2$ and there are four conics meeting two points and four lines. We solved 500 random real instances using the Brazos cluster. Each instance took an average of 0.7 seconds for Bertini to solve and 0.1 seconds for **alphaCertified** to certify the results. We found that there can be 0, 2, or 4 real solutions. Table 3 presents the frequency distribution of these 500 instances for this case.

Table 3. Frequency distribution for conics through two points and four lines

# real	0	2	4	total
frequency	12	221	267	500

When $k = 1$, there are 18 conics meeting a point and six lines in $\mathbb{C}^3$. We solved 1,000,000 random real instances using the Brazos cluster. Each instance took an average of 1.6 seconds for Bertini to solve and an average of 0.1 seconds for **alphaCertified** to certify the results. Every real instance that we computed had at least 2 real solutions. Table 4 presents the frequency distribution of these 1,000,000 instances for this case.

Table 4. Frequency distribution for conics through a point and six lines

# real	0	2	4	6	8	10	12	14	16	18	total
frequency	0	3281	21984	88813	193612	261733	226383	137074	53482	13638	1000000

To compare the performance of **alphaCertified** to symbolic methods, we computed 40,000 instances of the conic problem with $k = 1$ using Singular [11] to compute an eliminant that satisfies the Shape Lemma [7] and Maple to count the number of real roots of the eliminant, which is a standard symbolic method to determine the number of real solutions to a zero-dimensional system of polynomial equations. The coordinates of points were taken to be rational numbers p/q where p, q were integers with $|p| < 4000$ and $0 < q < 1000$. Each computation took approximately 661 seconds on a single node of a server with four six-core AMD Opteron 8435 processors and 64 GB of memory. Table 5 presents the frequency distribution of these 40,000 instances for this case.

Table 5. Frequency distribution for conics through a point and six lines

# real	0	2	4	6	8	10	12	14	16	18	total
frequency	0	146	892	3558	7739	10575	8965	5488	2089	548	40000

Finally, when $k = 0$, there are 92 plane conics meeting eight general lines in $\mathbb{C}^3$. We solved 15,662,000 random real instances using the Brazos cluster. On average, each instance took 8.8 seconds for Bertini to solve and 0.7 seconds for **alphaCertified** to certify the results. Table 6 presents the frequency distribution of these instances.

5.4. **A Schubert problem.** Our last example concerns a problem in the Schubert calculus of enumerative geometry, which is a rich class of geometric problems involving linear subspaces of a vector space. Many problems in the Schubert calculus are naturally formulated as overdetermined polynomial systems. We investigate one such problem that can also be formulated as a square polynomial system using the approach of [2]. In particular, we demonstrate **alphaCertified**'s algorithms for overdetermined systems as well as investigate a conjecture on the reality of its solutions.

This problem involves four-dimensional linear subspaces (four-planes) H of $\mathbb{C}^8$ that have a non-trivial intersection with each of eight general three-planes $K_0, \ldots, K_7$. The

Table 6. Frequency distribution for conics through eight lines

# real	0	2	4	6	8	10	12	14
frequency	1	8	26	65	466	1548	4765	11928
# real	16	18	20	22	24	26	28	30
frequency	26439	52875	98129	167932	270267	404918	569891	756527
# real	32	34	36	38	40	42	44	46
frequency	942674	1114033	1246533	1332289	1355320	1319699	1226667	1091019
# real	48	50	52	54	56	58	60	62
frequency	932838	762463	596174	449021	323927	223455	149629	95740
# real	64	66	68	70	72	74	76	78
frequency	59141	34834	19516	10672	5671	2744	1290	530
# real	80	82	84	86	88	90	92	total
frequency	204	90	26	11	3	2	0	15662000

Schubert calculus predicts 126 such four-planes. To formulate this Schubert problem, consider H to be the column space of a 8×4 matrix in block form

$$H \;=\; \left[\begin{array}{c} I_4 \\ X \end{array}\right],$$

where I_4 is the 4×4 identity matrix and X is a 4×4 matrix of indeterminates. Represent a three-plane K as the column space of a 8×3 matrix of constants. Then the condition that H meets K non-trivially is equivalent to the vanishing of the determinants of the eight 7×7 square submatrices of the 8×7 matrix

$$A \;=\; [H\ K]\,. \tag{9}$$

In this standard formulation, the Schubert problem is a system of 64 equations in 16 indeterminates. Using a total degree homotopy to solve this would follow 4^{16} paths.

There is a second formulation which we used. Write K in block form,

$$K \;=\; \left[\begin{array}{c} \mathcal{K}_1 \\ \mathcal{K}_2 \end{array}\right],$$

where $\mathcal{K}_1$ and $\mathcal{K}_2$ are 4×3 matrices. A linear dependency among the columns of A (9) is given by vectors $v \in \mathbb{C}^4$ and $w \in \mathbb{C}^3$ such that $Hv + Kw = 0$. Applying this to the different blocks of H and K gives

$$I_4 v + \mathcal{K}_1 w \;=\; 0 \qquad \text{and} \qquad Xv + \mathcal{K}_2 w \;=\; 0\,,$$

which is equivalent to $\widehat{A}w = 0$, where $\widehat{A} := \mathcal{K}_2 - X\mathcal{K}_1$. Thus H meets K non-trivially if and only if each 3×3 minor of $\widehat{A}$ vanishes. This gives a system $F_O(x)$ of 32 cubic polynomials in 16 indeterminates, which is more compact than the original formulation.

We certified solutions to this overdetermined polynomial system F_O. We randomized F_O to maintain the structure of the equations as follows. For each $i = 0, \ldots, 7$ and $j = 1, 2, 3, 4$, let $f_{i,j}$ be the determinant of the submatrix created by removing the j^{th} row of the matrix $\widehat{A}_i$ corresponding to the ith three-plane. Then, for each j, we take four random linear combinations of the polynomials $f_{0,j}, f_{1,j}, \ldots, f_{7,j}$. This preserves the multilinear structure of the equations in the four variable groups corresponding to the

columns of X. Solving this system using regeneration [21] finds 22,254 solutions of which **alphaCertified** soft certified using 256-bit precision that 126 of these are approximate solutions to two different random square subsystems of F_O with associated solutions within a distance of $\delta = 10^{-10}$ of each other. The same result was also obtained using $\delta = 10^{-5}$. Thus, **alphaCertified** provided a soft certificate based on the heuristic algorithm for overdetermined systems that we found all 126 solutions to the Schubert problem.

This Schubert problem has an equivalent formulation as a square system. The columns of $\widehat{A}$ are linearly dependent if and only if there exists $0 \neq v \in \mathbb{C}^3$ such that $\widehat{A}v = 0$. For generic $\alpha_1, \alpha_2 \in \mathbb{C}$, this occurs if and only if there exists $y_1, y_2 \in \mathbb{C}$ such that

$$\widehat{A} \cdot \left[\begin{array}{c} y_1 \\ y_2 \\ \alpha_1 y_1 + \alpha_2 y_2 + 1 \end{array} \right] \; = \; 0\,.$$

This yields a system of 32 polynomials in 32 indeterminates, say $F_S(x, y^{(0)}, \ldots, y^{(7)})$. This polynomial system consists of four bilinear polynomials in x and $y^{(i)}$ for each $i = 0, \ldots, 7$. Since $y^{(i)}$ consists of two indeterminates, namely $y_1^{(i)}$ and $y_2^{(i)}$, a 9-homogeneous homotopy used to solve F_S would follow $\binom{4}{2}^8 = 6^8$ paths. As described in [2], we are interested in the components of $\mathcal{V}(F_S)$ having fibers with generic dimension zero. For generic $K_0, \ldots, K_7$, since $\mathcal{V}(F_S)$ is zero-dimensional, $\mathcal{V}(F_O)$ and $\mathcal{V}(F_S)$ both consist of 126 isolated points and $\mathcal{V}(F_S)$ naturally projects onto $\mathcal{V}(F_O)$.

We also investigated the number of real solutions when the three planes K_i are as follows. For $t \in \mathbb{R}$, let $\gamma(t) = (1, t, t^2, \ldots, t^7) \in \mathbb{R}^8$ be a point on the moment curve. Select 24 rational numbers $t_1, \ldots, t_{24}$ and for $i = 0, \ldots, 7$, let K_i be the span of the three linearly independent vectors $\gamma(t_{3i+1})$, $\gamma(t_{3i+2})$, and $\gamma(t_{3i+3})$. When $t_1 < t_2 < \cdots < t_{24}$, the Secant Conjecture [16] posits that all 126 solutions will be real, but if the points are not in this or some equivalent order, then other numbers of real solutions are possible.

Since $K_0, \ldots, K_7$ are real, if the constants α_i are real, then the real points of $\mathcal{V}(F_O)$ correspond to the real points of $\mathcal{V}(F_S)$. We solved 25000 real instances using random numbers in the interval $[-1, 1]$ and softly certified that each had 126 real solutions, using 256-bit precision.

Lastly, we investigated the number of real solutions when the three planes K_i are as follows. For $i = 0, \ldots, 7$, let $t_i \in \mathbb{C}$ be generic under the condition that $2k$ are complex conjugate pairs and $8 - 2k$ are real, where $0 \leq k \leq 4$. Define $K_i = T(t_i)$ where

$$T(t) = \left[\begin{array}{ccc} 1 & 0 & 0 \\ t & 1 & 0 \\ t^2 & 2t & 1 \\ t^3 & 3t^2 & 3t \\ t^4 & 4t^3 & 6t^2 \\ t^5 & 5t^4 & 10t^3 \\ t^6 & 6t^5 & 15t^4 \end{array} \right].$$

Then K_i is the three-plane osculating the moment curve at the point $\gamma(t_i)$. When $k = 0$, that is, when each t_i is real, this is the Shapiro Conjecture (MTV Theorem) [39, 34]

and all 126 solutions are real. We tested 1000 such instances and for each, **alphaCertified** correctly identified all 126 solutions to be real. Our primary interest was when $k > 0$, for we wanted to test the hypothesis that there would be a lower bound to the number of real solutions if the set of osculating three-planes were real (that is, if $\{\overline{K_0}, \ldots, \overline{K_7}\} = \{K_0, \ldots, K_7\}$). This is what we found, as can be seen in the partial frequency table we give below. (To better show the lower bounds, we omit writing 0 in the cells with no observed instances.) This enumeration of real solutions was softly certified using 256-bit precision.

Table 7. Frequency distribution for the Schubert problem

	# real													
k	0	2	4	6	8	10	12	$\cdots$	18	20	22	$\cdots$	124	126	total
0								$\cdots$				$\cdots$		1000	1000
1				6	6	10	88	$\cdots$	554	1888	1832	$\cdots$	69	2021	42000
2						2614	3771	$\cdots$	3285	1579	1378	$\cdots$	1	38	24000
3						8896	4479	$\cdots$	1079	721	2586	$\cdots$			23500
4								$\cdots$			19134	$\cdots$		1	22500

This computation was part of a larger test of hypothesized lower bounds [20].

6. Conclusion

Smale's α-theory provides a way to certify solutions to polynomial systems, determine if two points correspond to distinct solutions, and determine if the corresponding solution is real. Using either exact rational or arbitrary precision floating point arithmetic, **alphaCertified** is a program which implements these α-theoretical methods.

We have also produced a Maple interface to **alphaCertified** to facilitate the construction of the input files needed.

Acknowledgements

The authors would like to thank Mike Shub for his helpful comments and the first author would like to thank the organizers of the Foundations of Computational Mathematics thematic program at the Fields Institute.

References

[1] E. Allgower and K. Georg, *Introduction to numerical continuation methods*, Classics in Applied Mathematics, 45, SIAM, 2003.

[2] D.J. Bates, J.D. Hauenstein, C. Peterson, and A.J. Sommese, *Numerical decomposition of the rank-deficiency set of a matrix of multivariate polynomials*, Approximate commutative algebra, Texts Monogr. Symbol. Comput., Springer, Vienna, 2009, pp. 55–77.

[3] D.J. Bates, J.D. Hauenstein, and A.J. Sommese, *Efficient path tracking methods*, Preprint, 2010.

[4] D.J. Bates, J.D. Hauenstein, A.J. Sommese, and C.W. Wampler, *Bertini: Software for numerical algebraic geometry*, Available at `http://www.nd.edu/~sommese/bertini`.

[5] D.J. Bates, J.D. Hauenstein, A.J. Sommese, and C.W. Wampler, II, *Adaptive multiprecision path tracking*, SIAM J. Numer. Anal. **46** (2008), no. 2, 722–746.

[6] ______, *Stepsize control for path tracking*, Interactions of classical and numerical algebraic geometry, Contemp. Math., vol. 496, Amer. Math. Soc., Providence, RI, 2009, pp. 21–31.
[7] E. Becker, M.G. Marinari, T. Mora, and C. Traverso, *The shape of the Shape Lemma*, Proceedings ISSAC-94, 1993, pp. 129–133.
[8] C. Beltrán and A. Leykin, *Certified numerical homotopy tracking*, `arXiv.org/0912.0920`, 2009.
[9] L. Blum, F. Cucker, M. Shub, and S. Smale, *Complexity and real computation*, Springer-Verlag, New York, 1998, With a foreword by Richard M. Karp.
[10] Brazos Computational Resource, *Academy for advanced telecommunications and learning technologies*, Texas A&M University.
[11] W. Decker, G.-M. Greuel, G. Pfister, and H. Schönemann, SINGULAR *3-1-1 — A computer algebra system for polynomial computations*, 2010, `http://www.singular.uni-kl.de`.
[12] J.-P. Dedieu and M. Shub, *Newton's method for overdetermined systems of equations*, Math. Comp. **69** (2000), no. 231, 1099–1115.
[13] P. Dietmaier, *The Stewart-Gough platform of general geometry can have 40 real postures*, Advances in Robot Kinematics: Analysis and Control, Kluwer Academic Publishers, 1998, pp. 1–10.
[14] L. Fousse, G. Hanrot, V. Lefèvre, P. Pélissier, and P. Zimmermann, *MPFR: a multiple-precision binary floating-point library with correct rounding*, ACM Trans. Math. Software **33** (2007), no. 2, Art. 13, 15.
[15] W. Fulton and R. Pandharipande, *Notes on stable maps and quantum cohomology*, Algebraic geometry—Santa Cruz 1995, Proc. Sympos. Pure Math., vol. 62, Amer. Math. Soc., Providence, RI, 1997, pp. 45–96.
[16] L. García-Puente, N. Hein, C. Hillar, A. Martín del Campo-Sanchez, J. Ruffo, F. Sottile, and Z. Teitler, *The Secant Conjecture in the real Schubert calculus*, `arXiv.org/1010.0665`, 2010.
[17] G.H. Golub and C.F. Van Loan, *Matrix computations*, third ed., Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, Baltimore, MD, 1996.
[18] V.E. Gough, *Contribution to discussion papers on research in automobile stability, control and tyre performance*, Proc. Auto Div. Inst. Mech. Eng. (1956–1957), 392–394.
[19] T. Granlund, *GNU MP: the gnu multiple precision arithmetic library*, Available at `http://www.gmplib.org`.
[20] J.D. Hauenstein, N. Hein, A. Martín del Campo-Sanchez, and F. Sottile, *Beyond the Shapiro Conjecture and Eremenko-Gabrielov lower bounds*, Available at `http://www.math.tamu.edu/~sottile/research/pages/lower_Shapiro/`.
[21] J.D. Hauenstein, A.J. Sommese, and C.W. Wampler, *Regeneration homotopies for solving systems of polynomials*, Math. Comp. **80** (2011), 345–377.
[22] J.D. Hauenstein and F. Sottile, *alphaCertified: Software for certifying numerical solutions to polynomial equations*, Available at `http://www.math.tamu.edu/~sottile/research/stories/alphaCertified`.
[23] I. Itenberg, V. Kharlamov, and E. Shustin, *A Caporaso-Harris type formula for Welschinger invariants of real toric del Pezzo surfaces*, Comment. Math. Helv. **84** (2009), no. 1, 87–126.
[24] G. Jeronimo and D. Perrucci, *On the minimum of a positive polynomial over the standard simplex*, J. Symbolic Comput. **45** (2010), no. 4, 434–442.
[25] E. Kostlan, *Random polynomials and the statistical fundamental theorem of algebra*, 1987, Available at `http://www.developmentserver.com/randompolynomials/rpsfta`.
[26] S. Lang, *Real analysis*, second ed., Addison-Wesley Publishing Company Advanced Book Program, Reading, MA, 1983.
[27] T.-L. Lee, T.-Y. Li, and C.-H. Tsai, *HOM4PS-2.0: A software package for solving polynomial systems by the polyhedral homotopy continuation method*, Computing **83** (2008), 109–133.
[28] A. Leykin, *NAG4M2: Numerical algebraic geometry for Macaulay 2*, Available at `http://people.math.gatech.edu/~aleykin3/NAG4M2`.
[29] A. Leykin and F. Sottile, *Galois groups of Schubert problems via homotopy computation*, Math. Comp. **78** (2009), no. 267, 1749–1765.

[30] G. Malajovich, *pss – Polynomial System Solver version 3.0.5*, Software available at http://www.labma.ufrj.br/~gregorio/software.php.

[31] G. Mikhalkin, *Enumerative tropical algebraic geometry in* $\mathbb{R}^2$, J. Amer. Math. Soc. **18** (2005), no. 2, 313–377 (electronic).

[32] A. Morgan, *Solving polynomial systems using continuation for engineering and scientific problems*, Classics in Applied Mathematics, 57, SIAM, 2009.

[33] A.P. Morgan, A.J. Sommese, and C.W. Wampler, *Complete solution of the nine-point path synthesis problem for four-bar linkages*, ASME J. Mech. Des. **114** (1992), no. 1, 153–159.

[34] E. Mukhin, V. Tarasov, and A. Varchenko, *The B. and M. Shapiro conjecture in real algebraic geometry and the Bethe ansatz*, Ann. of Math. (2) **170** (2009), no. 2, 863–881.

[35] M. Shub and S. Smale, *Complexity of Bézout's theorem. I. Geometric aspects*, J. Amer. Math. Soc. **6** (1993), no. 2, 459–501.

[36] S. Smale, *Newton's method estimates from data at one point*, The merging of disciplines: new directions in pure, applied, and computational mathematics (Laramie, Wyo., 1985), Springer, New York, 1986, pp. 185–196.

[37] A.J. Sommese, J. Verschelde, and C.W. Wampler, *Numerical irreducible decomposition using projections from points on the components*, Symbolic computation: solving equations in algebra, geometry, and engineering (South Hadley, MA, 2000), Contemp. Math., vol. 286, Amer. Math. Soc., Providence, RI, 2001, pp. 37–51.

[38] A.J. Sommese and C.W. Wampler, II, *The numerical solution of systems of polynomials arising in engineering and science*, World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, 2005.

[39] F. Sottile, *Frontiers of reality in Schubert calculus*, Bull. Amer. Math. Soc. (N.S.) **47** (2010), no. 1, 31–71.

[40] D. Stewart, *A platform with 6 degree of freedom*, Proc. of the Institution of Mechanical Engineers **180** (1965-66), 371–386.

[41] J. Verschelde, *Algorithm 795: PHCpack: A general-purpose solver for polynomial systems by homotopy continuation*, ACM Trans. Math. Softw. **25** (1999), no. 2, 251–276, Software available at http://www.math.uic.edu/~jan.

[42] ———, *Polynomial formulation and solutions of Stewart-Gough platform with 40 real positions*, 1999, http://www.math.uic.edu/~jan/Demo/stewgou40.html.

[43] J.-Y. Welschinger, *Invariants of real rational symplectic 4-manifolds and lower bounds in real enumerative geometry*, C. R. Math. Acad. Sci. Paris **336** (2003), no. 4, 341–344.

Jonathan D. Hauenstein, Department of Mathematics, Texas A&M University, College Station, Texas 77843, USA

E-mail address: jhauenst@math.tamu.edu

URL: http://www.math.tamu.edu/~jhauenst

Frank Sottile, Department of Mathematics, Texas A&M University, College Station, Texas 77843, USA

E-mail address: sottile@math.tamu.edu

URL: http://www.math.tamu.edu/~sottile